C# 프로그래밍

- 1 주차 (2강)

SEJONG UNIVERSITY

데이터 타입

- 변수
- 값 형식과 참조 형식
- 기본 데이터 형식

SEJONG UNIVERSITY

기본 용어 : 표현식, 문장

- 표현식 : 값을 만들어내는 간단한 코드

```
273
10 + 20 + 30 * 2
“C# Programming”
```

- 문장 (statement) :
  - ✓ 프로그래밍에서 실행 가능한 최소 단위의 코드
  - ✓ “;” 기호는 문장의 끝을 의미

```
273;
10 + 20 + 30 * 2;
Console.Write(“C# Programming”);
```

기본 용어 : 키워드

- 키워드 :
  - ✓ C# 언어가 미리 선택하여 의미부여한 단어
  - ✓ 일반 키워드는 개발자가 변수로 선언 불가
  - ✓ 문맥 키워드는 특정 위치에서만 키워드로 동작

일반 키워드

bool	break	class	Const
continue	delegate	do	else
event	extern	float	for
goto	long	struct	switch
…	…	…	…

문맥 키워드

add	alias	ascending	async
by	equals	from	global
…	…	…	…

기본 용어 : 식별자

- 식별자 : 어떤 대상을 유일하게 구별 가능하게 하는 이름
  - ✓ 키워드 사용 불가
  - ✓ 특수문자는 "_"만 허용
  - ✓ 숫자로 시작 불가
  - ✓ 공백 입력 불가

허용된 변수 또는 메서드 이름

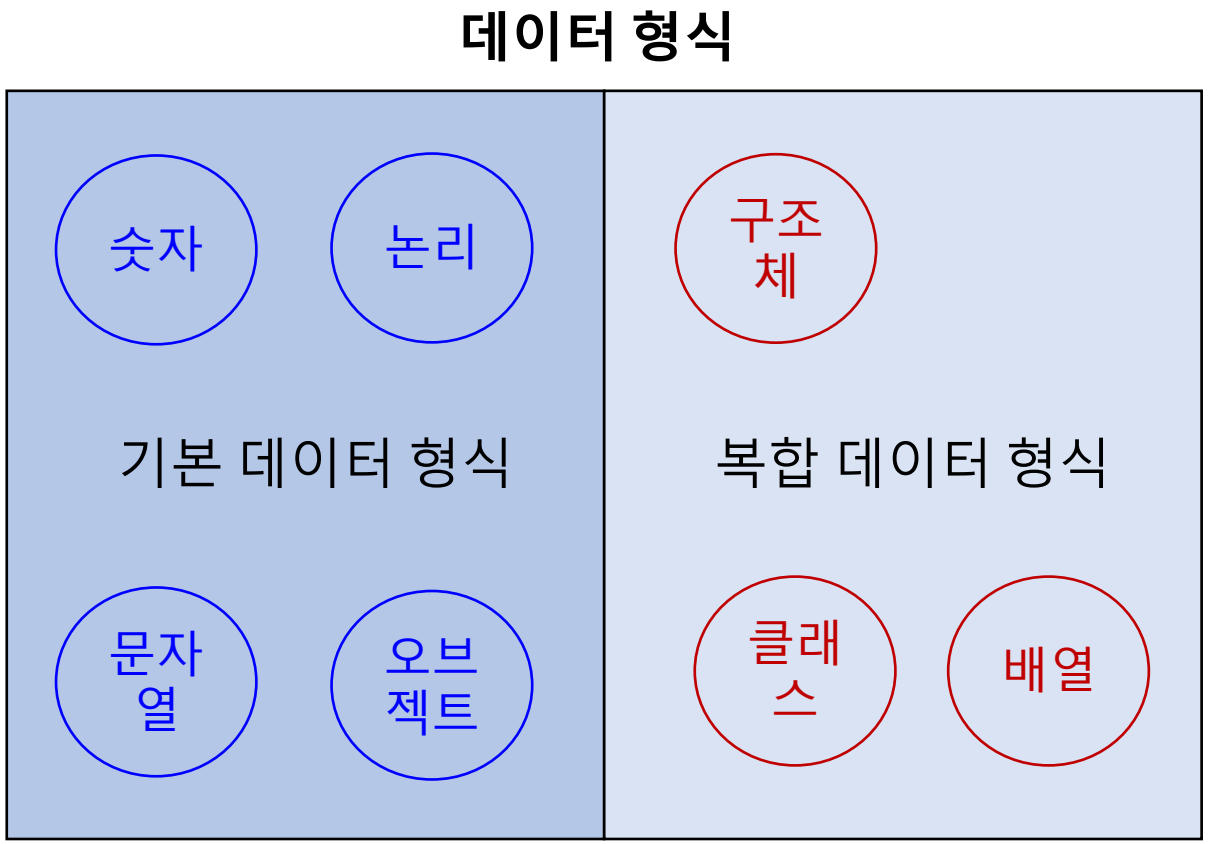
```
alpha
alpha10
_alpha
Alpha
ALPHA
```

에러 발생

```
break
273alpha
has space
```

C# 언어의 데이터 종류

- 기본 데이터 형식
  - ✓ 모든 데이터의 근간
  - ✓ 총 15가지 존재
  - ✓ 크게 숫자, 논리, 문자열, 오브젝트 형식으로 분류
- 복합 데이터 형식
  - ✓ 이미지나 소리 등의 데이터 표현
  - ✓ 구조체, 클래스, 배열 포함



데이터 타입

- 변수

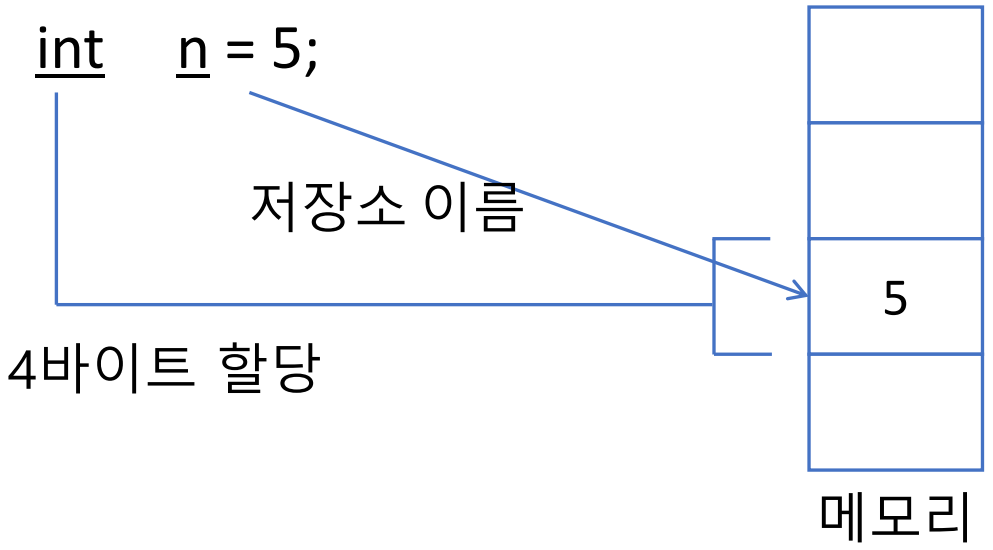
- 값 형식과 참조 형식

- 기본 데이터 형식

변수의 선언

- 변수 선언
  - ✓ 컴파일러는 변수를 위한 메모리 할당
  - ✓ 대입 연산자 "="를 통해 값 입력
  - ✓ 초기화 되지 않은 변수 사용시 컴파일 에러 발생

```
int x;
Console.Write(x);  // 컴파일 에러 발생
```



  - ✓ 다수의 변수 동시 선언 가능

```
int a, b, c;                    // 같은 형식의 변수
int x = 30, y = 40, z = 50 ;    // 선언과 초기화를 한번에 진행
```

리터럴 (Literal)

- 사전적 의미 "문자 그대로의"
- 프로그램 언어에서는 "고정 값을 나타내는 표기법"
- 예: "int x = 30"
  - ✓ x는 변수이고, 30은 리터럴
  - ✓ x는 입력 값에 따라 변하지만, 30은 보이는 그대로의 고정 값

```
Int      a = 100;                  // 리터럴: 100 (정수형)
Int      b = 0x200;                // 리터럴: 0x200 (16진수 표기, 10진수의 512)
float    c = 3.14f;                // 리터럴: 3.14f (float형 표기)
double   d = 0.12345678;           // 리터럴: 0.12345678 (double형)
string   s = "가나다라마바사";      // 리터럴: "가나다라마바사" (문자열 형)
```

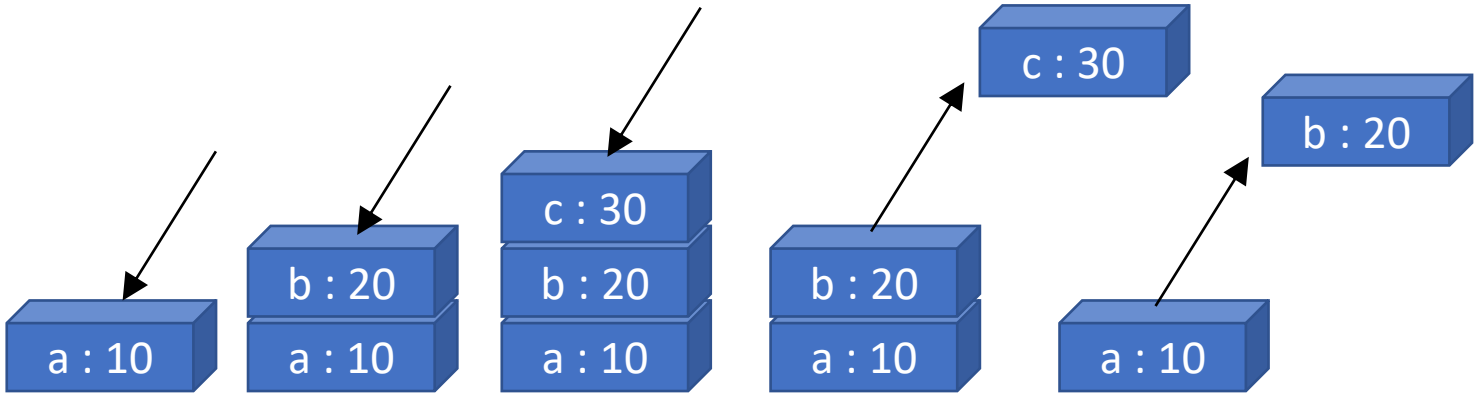
데이터 타입

- 변수
- 값 형식과 참조 형식
- 기본 데이터 형식

스택 메모리

- 데이터를 탑을 쌓 듯이 밑부터 순차적으로 쌓고 꺼낼 때는 위에서부터 순차적으로 꺼낼 수 있는 구조
- 예를 들어, (아래 박스 안에 코드의 경우)
  - ✓ 코드 블록이 시작하는 "{" 부터 변수 a, b, c 가 순차적으로 쌓임
  - ✓ 코드 끝 "}"부터 c. b. a의 순으로 메모리가 비워 짐
- 스택은 코드 끝에서 메모리 자동제거
- 값 형식의 데이터 스택 메모리 사용

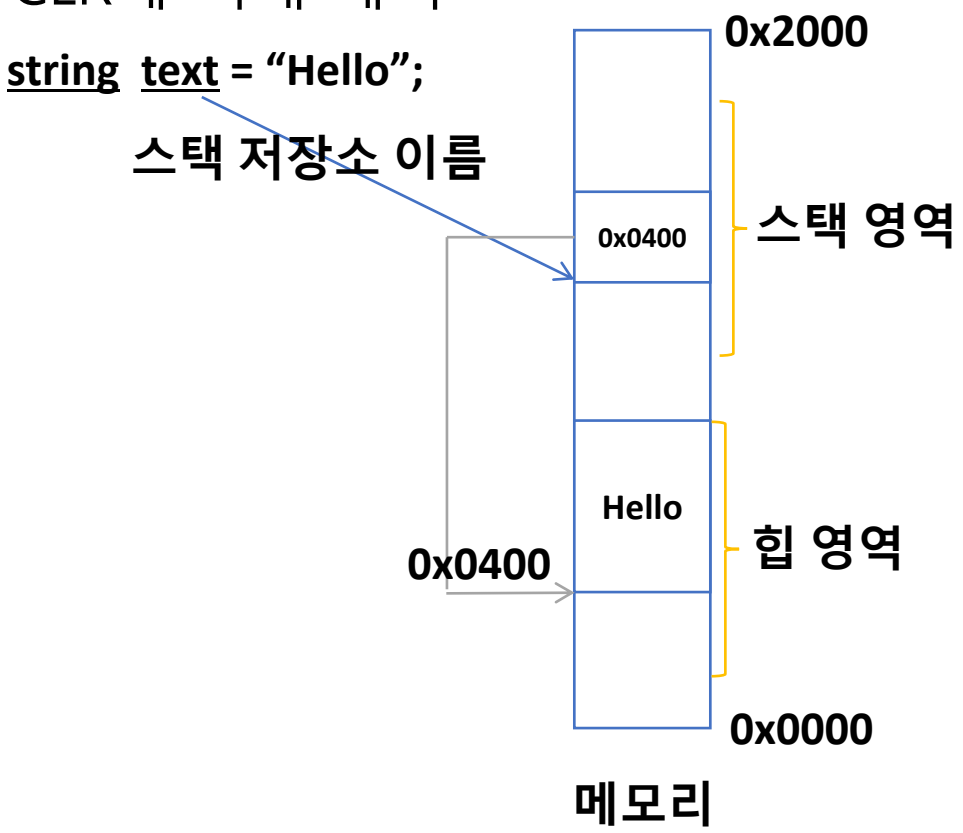
```
{ // 코드 블록 시작
        int a = 100;
        int b = 200;
        int c = 300;
} // 코드 블록 끝
```



힙 메모리

- 메모리 제거를 위해 CLR의 가비지 컬렉터 이용
- 개발자가 원할 때까지 데이터를 메모리에 유지 가능
  - ✓ 스택은 메모리 자동 비움
  - ✓ 나중에 데이터를 앞으로 사용할 일이 없는 경우 CLR에 의해 제거
- 참조 형식은 스택과 힙 메모리 모두 사용
  - ✓ 힙 영역 : 데이터를 저장
  - ✓ 스택 영역 : 데이터가 저장된 힙 메모리 주소

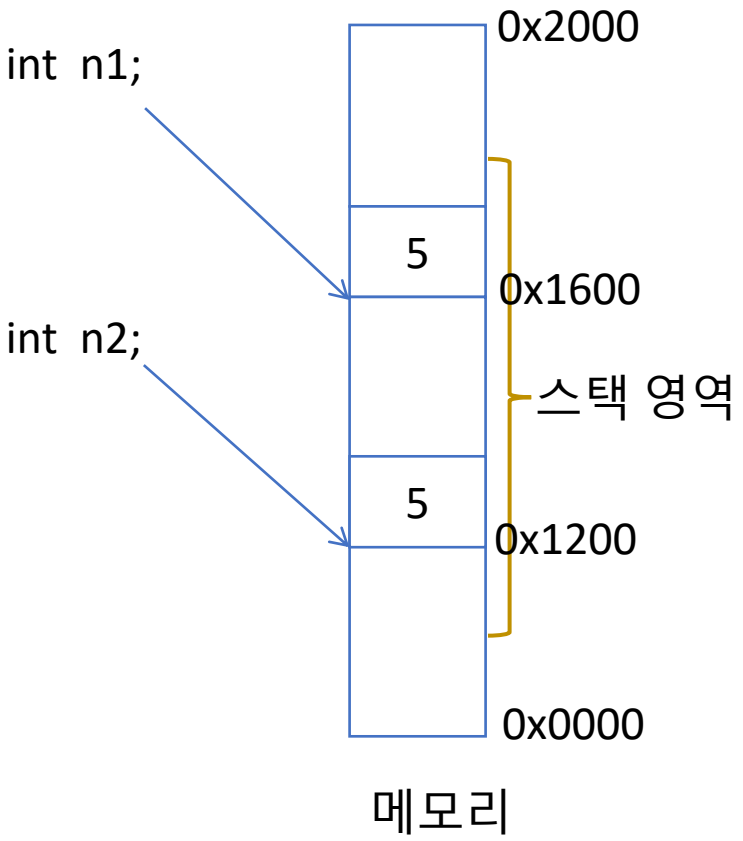
```
{
        string text = “Hello”;
}
```

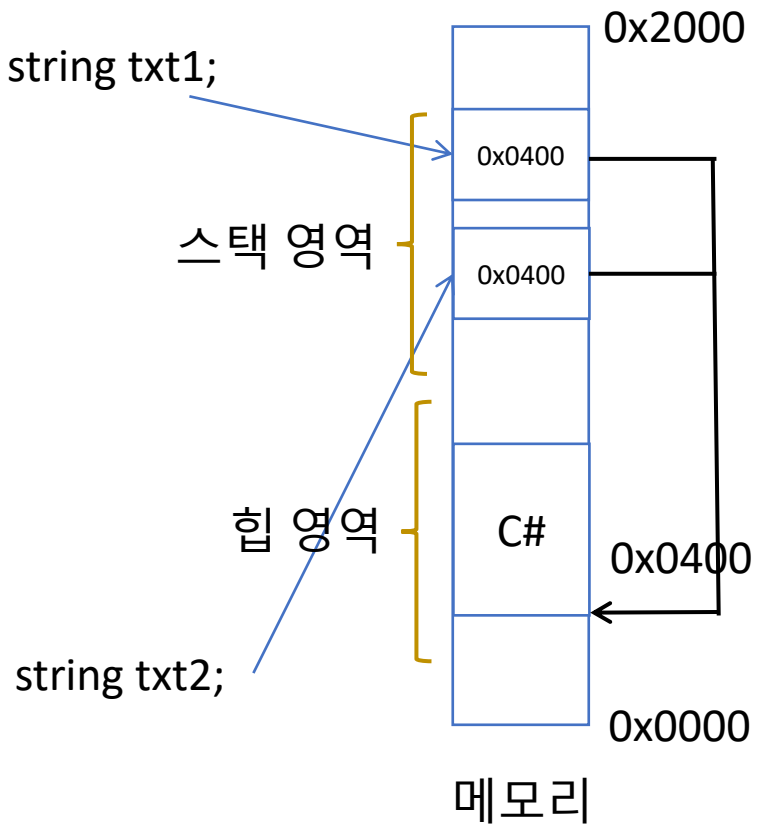


값 형식과 참조 형식 차이점 (예제 코드)

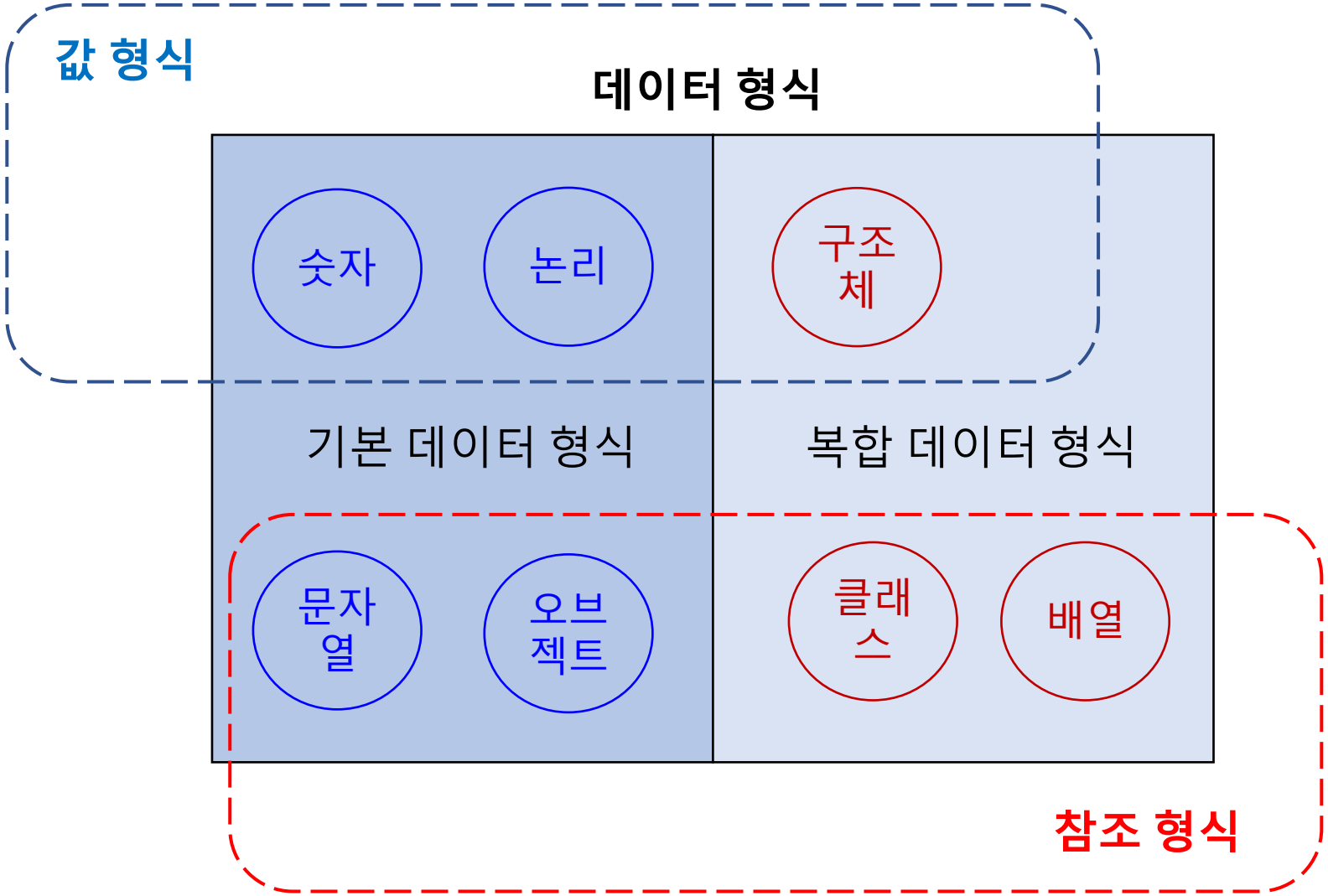
```
{
    int n1 = 5;
    int n2 = n1;

    string txt1 = "C#";
    string txt2 = txt1;
}
```





C# 데이터 형 분류



데이터 타입

- 변수
- 값 형식과 참조 형식
- 기본 데이터 형식

숫자 데이터 형식 : 정수형

- C#의 숫자 데이터형식은 12 가지
  - ✓ 정수 계열, 부동 소수 계열, 소수 계열
- 정수 계열 형식
  - ✓ 정수 데이터를 담기 위한 형식
  - ✓ 형식 크기와 값 범위가 다른 총 9가지
- 코드에 사용될 데이터의 범위를 예측한 후 적절한 데이터형 선택

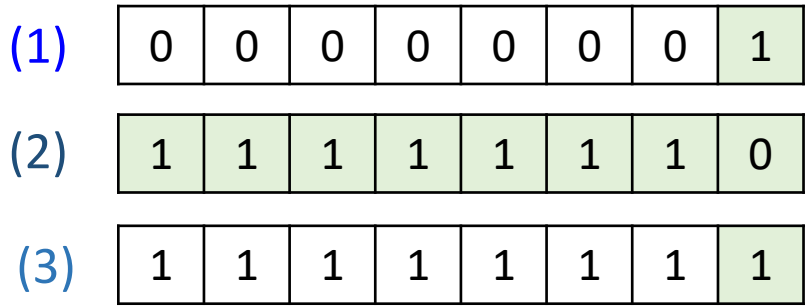
형식	크기 (바이트)	값의 범위
byte	1	0~255
sbyte	1	-128~127
short	2	-32,768~32,767
ushort	2	0~65,535
int	4	--2,147,483,648 ~ 2,147,483,647
uint	4	…
long	8	…
ulong	8	…
Char	2	

숫자 데이터 형식 : 정수형

- 2/10/16 진수 리터럴 구분을 위해 접두사 지원
  - ✓ 2진수 리터럴을 위해 “0b”, 예: 0b1111_0000 (10진수 240);
  - ✓ 16진수 리터럴을 위해 “0x”, 예: 0xF0 (10진수 240);
  - ✓ 접두사가 붙지 않은 경우 10진수로 간주

- 부호 있는 정수와 부호 없는 정수
  - ✓ 부호 없는 정수 byte, ushort, uint, ulong
  - ✓ 부호는 간단하게 (+) 또는 (-)를 나타냄
  - ✓ 2의 보수법으로 음수 표현

2의 보수법 (예: -1)



(1) 1을 수 부분에 입력

(2) 1은 0으로 0은 1로 반전

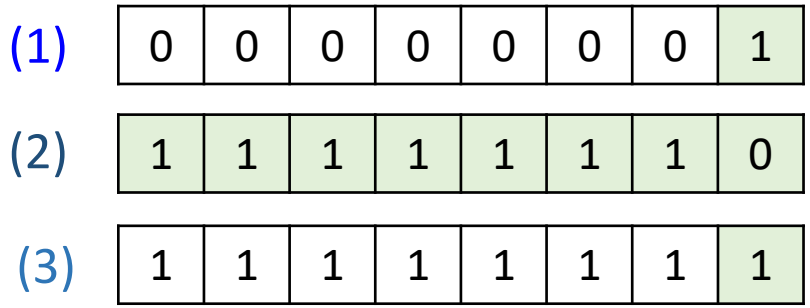
(3) 반전된 비트에 1을 더함

숫자 데이터 형식 : 정수형

- 2/10/16 진수 리터럴 구분을 위해 접두사 지원
  - ✓ 2진수 리터럴을 위해 “0b”, 예: 0b1111_0000 (10진수 240);
  - ✓ 16진수 리터럴을 위해 “0x”, 예: 0xF0 (10진수 240);
  - ✓ 접두사가 붙지 않은 경우 10진수로 간주

- 부호 있는 정수와 부호 없는 정수
  - ✓ 부호 없는 정수 byte, ushort, uint, ulong
  - ✓ 부호는 간단하게 (+) 또는 (-)를 나타냄
  - ✓ 2의 보수법으로 음수 표현

2의 보수법 (예: -1)



(1) 1을 수 부분에 입력

(2) 1은 0으로 0은 1로 반전

(3) 반전된 비트에 1을 더함

숫자 데이터 형식 : 정수형

- 오버플로 (Overflow)
  - ✓ 변수의 데이터 형식 크기를 넘어선 값을 입력
  - ✓ 예: uint 형의 최대값에 1을 더한 값
  - ✓ uint 형의 최대값은 4,294,967,295
    이지만 결과는 의도치 않게 출력

```
{
    uint a = uint.MaxValue;
    a = a + 1;
    Console.WriteLine (a);
}  // 결과는 0을 출력
```

- 언더플로 (Underflow)
  - ✓ 변수의 최저 값보다 작은 데이터 저장
  - ✓ byte 형식 변수에 -1를 저장 하면 변수 결과값은 255

Thank you!